

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	David E. Lowell	§ Art Unit:	2195
		§	
Serial No.:	10/676,922	§ Confirmation No.:	8294
		§	
Filed:	October 1, 2003	§ Examiner:	Eric Charles Wai
		§	
For:	Runtime Virtualization and Devirtualization of I/O Devices by a Virtual Machine Monitor	§ Atty. Dkt. No.:	200309154-1 (HPC.0518US)
		§	

Mail Stop Amendment

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUMMARY OF TELEPHONIC INTERVIEW

Sir:

On June 24, 2010, Examiner Eric Wai called the undersigned to indicate that the Examiner will provide amended claims for consideration by Applicant to place the application in condition for allowance. On June 25, 2010, Examiner Wai sent an email to the undersigned containing the proposed amendments to claims 11, 15, and 17. After further discussion on July 1, 2010, the undersigned sent proposed amendments modified from the Examiner's proposals to the Examiner by email. The proposed amendments by the undersigned is reflected in the Unofficial Amendment attached herewith. Agreement was reached to place the case in condition for allowance.

The Examiner indicated that the Examiner has found further art, including U.S. Patent No. 6,968,398 and U.S. Patent Publication No. 2003/0097393. It is the Examiner's position that these references "teach the invention as currently claimed," *i.e.*, as claimed prior to the amendments made in the Unofficial Amendment attached herewith. Since Applicant has not had an opportunity to consider the art in view of the invention as claimed prior to the present amendments, Applicant disagrees that the Examiner's assertions are correct.

Initially, the Examiner also indicated that terminal disclaimers should be submitted with respect to Serial Nos. 10/677,159 and 10/676,577. However, on July 1,

2010, the Examiner indicated that the Examiner has spoken with the Examiner's Supervisor, and in view of the fact that those other two cases remain under rejection, terminal disclaimers are not required in the present case.

Agreement was reached.

Respectfully submitted,

Date: July 2, 2010

/Dan C. Hu/

Dan C. Hu
Registration No. 40,025
TROP, PRUNER & HU, P.C.
1616 South Voss Road, Suite 750
Houston, TX 77057-2631
Telephone: (713) 468-8880
Facsimile: (713) 468-8883

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	David E. Lowell	§ Art Unit:	2195
		§	
Serial No.:	10/676,922	§ Confirmation No.:	8294
		§	
Filed:	October 1, 2003	§ Examiner:	Eric Charles Wai
		§	
For:	Runtime Virtualization and Devirtualization of I/O Devices by a Virtual Machine Monitor	§ Atty. Dkt. No.:	200309154-1 (HPC.0518US)
		§	

Mail Stop Amendment

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

UNOFFICIAL AMENDMENT

Sir:

Please amend the above-identified application as follows:

Amendments to the Claims are reflected in the listing of claims, which begins on page 2 of this paper.

Remarks/Arguments begin on page 14 of this paper.

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1 1. (Cancelled)

1 2. (Previously Presented) The method of claim 11, wherein the computer further
2 includes a CPU, wherein the virtual machine monitor is in control of the CPU prior to the
3 runtime virtualization of the I/O device.

1 3. (Previously Presented) The method of claim 11, wherein the virtualization is
2 performed transparently to an operating system.

1 4. (Previously Presented) The method of claim 11, wherein the I/O device is
2 compatible with the virtualized I/O device.

1 5. (Cancelled)

1 6. (Previously Presented) The method of claim 11, further comprising configuring
2 hardware to trap I/O accesses, and enabling the virtual machine monitor to emulate the I/O
3 device in response to the trapped I/O accesses.

1 7. (Original) The method of claim 6, wherein the virtual machine monitor uses
2 memory management to trap the I/O accesses.

1 8. (Previously Presented) The method of claim 15, wherein the virtual machine
2 monitor commences the virtualization between I/O sequences.

1 9. (Currently Amended) The method of claim 8, wherein the virtual machine
2 monitor commences the virtualization by intercepting I/O accesses; wherein the virtual machine
3 monitor uses the intercepted I/O accesses to update [[a]]the state machine, whereby the state
4 machine reflects a state of the I/O device; and wherein the virtual machine monitor examines
5 transitions in the state machine to determine whether the I/O device is in the middle of an I/O
6 sequence.

1 10. (Previously Presented) The method of claim 15, further comprising the virtual
2 machine monitor commencing the virtualization during an I/O sequence.

1 11. (Currently Amended) In a computer including an I/O device, a method
2 comprising using a virtual machine monitor to commence virtualization of the I/O device at
3 runtime, wherein runtime is a period of execution in the computer after boot and before
4 shutdown of the computer, wherein virtualization of the I/O device is performed by:
5 redirecting interrupts for the I/O device to interrupt handlers in the virtual
6 machine monitor;
7 enabling the virtual machine monitor to intercept I/O accesses by an operating
8 system;
9 commencing emulation of the I/O device using a state machine;
10 wherein the virtual machine monitor determines whether the I/O device is performing an
11 I/O sequence, and delays commencing the virtualization until the virtual machine monitor
12 determines that the I/O sequence has completed.

1 12. (Previously Presented) The method of claim 11, wherein the runtime
2 virtualization includes using the virtual machine monitor to emulate I/O device interrupts.

1 13. (Currently Amended) The method of claim 11, wherein I/O device interrupts are
2 directed to [[an]]the operating system prior to the runtime virtualization of the I/O device; and
3 wherein the I/O device interrupts are directed to the virtual machine monitor during and after the
4 virtualization of the I/O device.

1 14. (Currently Amended) The method of claim 11, wherein the virtual machine
2 monitor temporarily pauses [[an]]~~the~~ I/O sequence by emulating the I/O device as being busy.

1 15. (Currently Amended) In a computer including an I/O device, a method
2 comprising:
3 using a virtual machine monitor to commence virtualization of the I/O device at runtime,
4 wherein runtime is a period of execution in the computer after boot and before shutdown of the
5 computer, wherein the I/O device has multiple modes of operations;
6 the virtual machine monitor determining the mode of the I/O device prior to commencing
7 the virtualization; and
8 the virtual machine monitor restoring the determined mode of operation after the
9 virtualization, wherein virtualization of the I/O device is performed by:
10 redirecting interrupts for the I/O device to interrupt handlers in the virtual
11 machine monitor;
12 enabling the virtual machine monitor to intercept I/O accesses by an operating
13 system; and
14 commencing emulation of the I/O device using a state machine.

1 16. (Previously Presented) The method of claim 11, further comprising devirtualizing
2 the I/O device at runtime following the runtime virtualization.

1 17. (Currently Amended) In a computer including hardware, a method comprising:
2 running a virtual machine monitor on the hardware;
3 running an operating system on the virtual machine monitor,
4 wherein the hardware includes an I/O device, and the I/O device is already virtualized by
5 the virtual machine monitor; and

6 devirtualizing the I/O device at runtime, wherein runtime is a period of execution in the
7 computer after boot and before shutdown of the computer, wherein devirtualizing the I/O device
8 is performed by:

9 waiting for a state machine for the I/O device to reach a particular state and
10 blocking new I/O requests;

11 draining I/O requests queued by the virtual machine monitor; and

12 redirecting interrupts for the I/O device to interrupt handlers in the operating
13 system.

1 18. (Original) The method of claim 17, wherein the devirtualization is performed
2 transparently to the operating system.

1 19. (Original) The method of claim 17, wherein the devirtualization includes stopping
2 I/O device emulation at runtime.

1 20. (Original) The method of claim 17, wherein the virtual machine monitor emulates
2 the I/O device prior to devirtualization; and wherein the devirtualization includes allowing the
3 virtual machine monitor to temporarily stop the operating system from commencing a new I/O
4 sequence.

1 21. (Original) The method of claim 20, wherein the virtual machine monitor
2 temporarily stops the operating system by emulating the I/O device as being in a "busy" or
3 "device not ready" state.

1 22. (Original) The method of claim 20, wherein the virtual machine monitor bounds
2 the amount of time the operating system processing is temporarily stopped.

1 23. (Previously Presented) The method of claim 20, further comprising:
2 the virtual machine monitor logging I/O accesses by the operating system to the I/O
3 device during devirtualization, and
4 replaying the log to the I/O device after devirtualization, wherein the I/O accesses by the
5 operating system are deferred during the devirtualization of the I/O device.

1 24. (Original) The method of claim 17, wherein the virtual machine monitor waits for
2 I/Os initiated by the virtual machine monitor's driver for the I/O device to complete, and for all
3 expected interrupts from the device to arrive, before ceasing device emulation.

1 25. (Cancelled)

1 26. (Previously Presented) The method of claim 17, further comprising, after
2 performing the devirtualizing, configuring the hardware so accesses by the operating system to
3 the I/O device no longer trap to the virtual machine monitor.

1 27. (Previously Presented) The method of claim 17, wherein the I/O device has
2 multiple modes of operations, the method further comprising:
3 the virtual machine monitor determining the mode of the I/O device prior to commencing
4 the devirtualization; and
5 the virtual machine monitor restoring the determined mode of operation after
6 devirtualization.

1 28. (Previously Presented) The method of claim 17, further comprising virtualizing
2 the I/O device at runtime again after performing the devirtualizing at runtime.

1 29. (Cancelled)

1 30. (Previously Presented) The computer of claim 35, wherein the I/O device is
2 compatible with the virtualized I/O device.

1 31. (Cancelled)

1 32. (Previously Presented) The computer of claim 35, wherein the hardware is
2 configured to trap I/O accesses, and the virtual machine monitor is enabled to emulate the I/O
3 device in response to the trapped I/O accesses.

1 33. (Previously Presented) The computer of claim 32, wherein the virtual machine
2 monitor is configured to use memory management to trap the I/O accesses.

1 34. (Cancelled)

1 35. (Currently Amended) A computer comprising:
2 hardware including an I/O device; and
3 computer memory encoded with a virtual machine monitor for running on the hardware
4 and commencing virtualization of the I/O device at runtime, wherein runtime is a period of
5 execution in the computer after boot and before shutdown of the computer, wherein the
6 virtualization of the I/O device is performed by:
7 redirecting interrupts for the I/O device to interrupt handlers in the virtual
8 machine monitor;
9 enabling the virtual machine monitor to intercept I/O accesses by an operating
10 system;
11 commencing emulation of the I/O device using a state machine,
12 wherein the virtual machine monitor is configured to determine whether the I/O device is
13 performing an I/O sequence, and to delay commencing the virtualization until the virtual
14 machine monitor determines that the I/O sequence has completed.

1 36. (Previously Presented) The computer of claim 35, wherein the virtual machine
2 monitor is configured to temporarily pause the I/O sequence by emulating the I/O device as
3 being busy.

1 37. (Previously Presented) The computer of claim 35, wherein the runtime
2 virtualization includes using the virtual machine monitor to emulate I/O device interrupts.

1 38. (Currently Amended) A computer comprising:
2 hardware including an I/O device; and
3 computer memory encoded with a virtual machine monitor for devirtualizing the I/O
4 device at runtime, wherein runtime is a period of execution in the computer after boot and before
5 shutdown of the computer, wherein devirtualization of the I/O device is performed by:
6 waiting for a state machine for the I/O device to reach a particular state and
7 blocking new I/O requests;
8 draining I/O requests queued by the virtual machine monitor; and
9 redirecting interrupts for the I/O device to interrupt handlers in the operating
10 system.

1 39. (Previously Presented) The computer of claim 38, wherein the virtual machine
2 monitor is configured to emulate the I/O device prior to commencing the devirtualization; and
3 wherein the virtual machine is configured to commence the devirtualization by temporarily
4 stopping an operating system running on the virtual machine monitor from commencing a new
5 I/O sequence.

1 40. (Previously Presented) The computer of claim 39, wherein the virtual machine
2 monitor is configured to temporarily stop the operating system by emulating the I/O device as
3 being in a "busy" or "device not ready" state.

1 41. (Previously Presented) The computer of claim 39, wherein the virtual machine
2 monitor is configured to bound the amount of time the operating system processing is
3 temporarily stopped.

1 42. (Previously Presented) The computer of claim 38, wherein the virtual machine
2 monitor is configured to log I/O accesses by an operating system to the I/O device during
3 devirtualization, and to replay the log to the I/O device after devirtualization.

1 43. (Previously Presented) The computer of claim 39, wherein the virtual machine
2 monitor is configured to wait for I/Os initiated by a virtual machine monitor driver for the I/O
3 device to complete, and for all expected interrupts from the I/O device to arrive, before ceasing
4 device emulation.

1 44. (Previously Presented) The computer of claim 38, wherein the hardware is
2 configured so operating system accesses to the I/O device no longer trap to the virtual machine
3 monitor after the devirtualization.

1 45. (Previously Presented) The computer of claim 38, wherein the I/O device has
2 multiple modes of operations; wherein the virtual machine monitor is configured to determine
3 the mode of the I/O device prior to commencing the devirtualization; and wherein the virtual
4 machine monitor is configured to restore the determined mode of operation after the I/O device
5 has been devirtualized.

1 46. (Previously Presented) The computer of claim 38, wherein the virtual machine
2 monitor is configured to further virtualize the I/O device after having devirtualized the I/O
3 device at runtime.

1 47. (Cancelled)

1 48. (Previously Presented) The article of claim 52, wherein the virtualization includes
2 commencing I/O device emulation at runtime.

1 49. (Previously Presented) The article of claim 48, wherein the virtual machine
2 monitor configures the hardware to trap I/O accesses, and enables the virtual machine monitor to
3 emulate the I/O device in response to the trapped I/O devices.

1 50. (Previously Presented) The article of claim 49, wherein the virtual machine
2 monitor uses memory management to trap the I/O accesses.

1 51. (Cancelled)

1 52. (Currently Amended) An article for a computer including an I/O device, the
2 article comprising computer-readable memory encoded with a virtual machine monitor for
3 causing the computer to commence virtualization of the I/O device at runtime, wherein runtime
4 is a period of execution in the computer after boot and before shutdown of the computer, wherein
5 virtualization of the I/O device is performed by:

6 redirecting interrupts for the I/O device to interrupt handlers in the virtual
7 machine monitor;

8 enabling the virtual machine monitor to intercept I/O accesses by an operating
9 system;

10 commencing emulation of the I/O device using a state machine,
11 wherein the virtual machine monitor determines whether the I/O device is performing an
12 I/O sequence, the virtual machine monitor delaying the commencement of the virtualization until
13 the virtual machine monitor determines that the I/O sequence has completed.

1 53. (Original) The article of claim 52, wherein the virtual machine monitor
2 temporarily pauses the I/O sequence by emulating the I/O device as being busy.

1 54. (Previously Presented) The article of claim 52, wherein the virtual machine
2 monitor emulates I/O device interrupts during the runtime virtualization.

1 55. (Cancelled)

1 56. (Currently Amended) An article for a computer including an I/O device, the
2 article comprising computer-readable memory encoded with a virtual machine monitor for
3 causing the computer to devirtualize the I/O device at runtime, wherein runtime is a period of
4 execution in the computer after boot and before shutdown of the computer, wherein
5 devirtualizing the I/O device is performed by:

6 waiting for a state machine for the I/O device to reach a particular state and
7 blocking new I/O requests;
8 draining I/O requests queued by the virtual machine monitor; and
9 redirecting interrupts for the I/O device to interrupt handlers in the operating
10 system.

1 57. (Original) The article of claim 56, wherein the devirtualization includes ceasing
2 emulation of the I/O device at runtime.

1 58. (Previously Presented) The article of claim 57, wherein the devirtualization
2 includes temporarily stopping an operating system running on the virtual machine monitor from
3 commencing a new I/O sequence.

1 59. (Original) The article of claim 58, wherein the virtual machine monitor
2 temporarily stops the operating system by emulating the I/O device as being in a "busy" or
3 "device not ready" state.

1 60. (Original) The article of claim 58, wherein the virtual machine monitor bounds
2 the amount of time the operating system processing is temporarily stopped.

1 61. (Previously Presented) The article of claim 57, wherein the virtual machine
2 monitor waits for I/Os initiated by a virtual machine monitor driver for the I/O device to
3 complete, and for all expected interrupts from the I/O device to arrive, before ceasing device
4 emulation.

1 62. (Previously Presented) The article of claim 56, wherein the virtual machine
2 monitor logs I/O accesses by an operating system to the I/O device during devirtualization, and
3 replays the log to the I/O device after devirtualization.

1 63. (Previously Presented) The article of claim 56, wherein the virtual machine
2 monitor, configures the hardware so operating system accesses to the I/O device do not trap to
3 the virtual machine monitor.

1 64. (Previously Presented) The article of claim 56, wherein the I/O device has
2 multiple modes of operations; and wherein the virtual machine monitor determines the mode of
3 the I/O device prior to commencing devirtualization; and restore the determined mode of
4 operation after the I/O device has been devirtualized.

1 65. (Cancelled)

1 66. (Previously Presented) The article of claim 56, wherein the virtual machine
2 monitor causes the computer to further virtualize the I/O device after having devirtualized the
3 I/O device at runtime.

REMARKS

As discussed.

Respectfully submitted,

Date: _____

//

Dan C. Hu
Registration No. 40,025
TROP, PRUNER & HU, P.C.
1616 South Voss Road, Suite 750
Houston, TX 77057-2631
Telephone: (713) 468-8880
Facsimile: (713) 468-8883